IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marphil Mokoena
2022/02/22

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

Summary of all results

- Exploratory Data Analysis result
- Interactive analytics in screenshots
- Predictive Analytics result

Introduction

Project background and context

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch.

Problems you want to find answers

- What factors determine if the rocket will land successfully?
- Which interactions amongst various features determine the success rate of a successful landing?
- What operating conditions needs to be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology**
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- **Perform data wrangling**
  - One-hot encoding data fields for Machine Learning and dropping irrelevant columns
- **Perform exploratory data analysis (EDA) using visualization and SQL**
  - Scatter and Bar Graphs to reveal patterns amongst data sets
- **Perform interactive visual analytics**
  - Folium and Plotly Dash Visualisations
- **Perform predictive analysis using classification models**
  - Building, tuning, and evaluating classification models

Data Collection

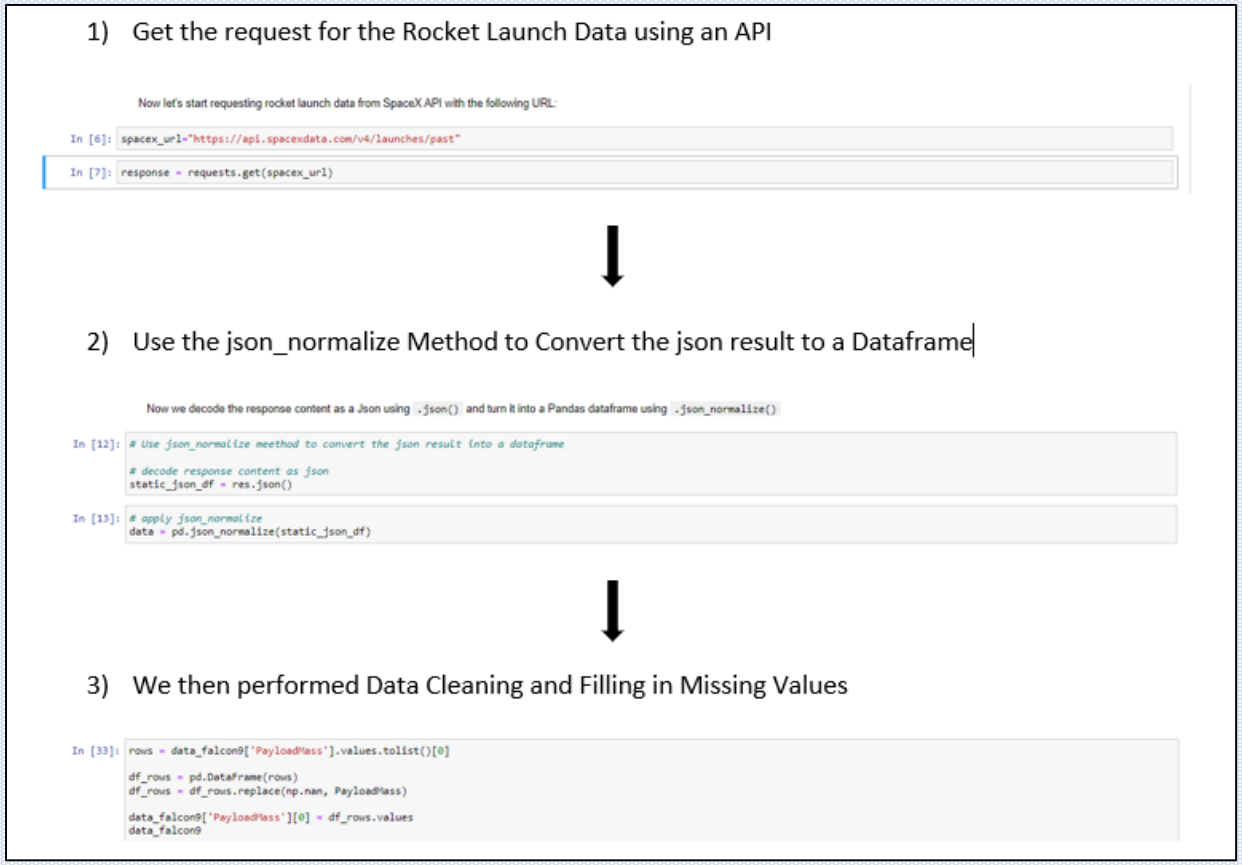
❖ **The data was collected using various methods**

- Data collection was done using GET Request on the SpaceX API.
- The response content was decoded as a json using the .json() function call and was turned into a pandas dataframe using .json_normalize().
- Data was cleaned by searching for missing values and filling in missing values.
- Web Scraping was performed from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- Launch Records were extracted as a HTML table which was parsed and converted into a pandas dataframe.

Data Collection – SpaceX API

- GET Request was used on the SpaceX API to collect data. After cleaning the data, basic data wrangling and formatting was conducted.

- GitHub URL: https://github.com/MarphilMokoena/Space-X/blob/56074c6dac57857a9d93104f5ceb3bbadaac7b57/Data%20Collection%20%E2%80%93%20SpaceX%20API.ipynb



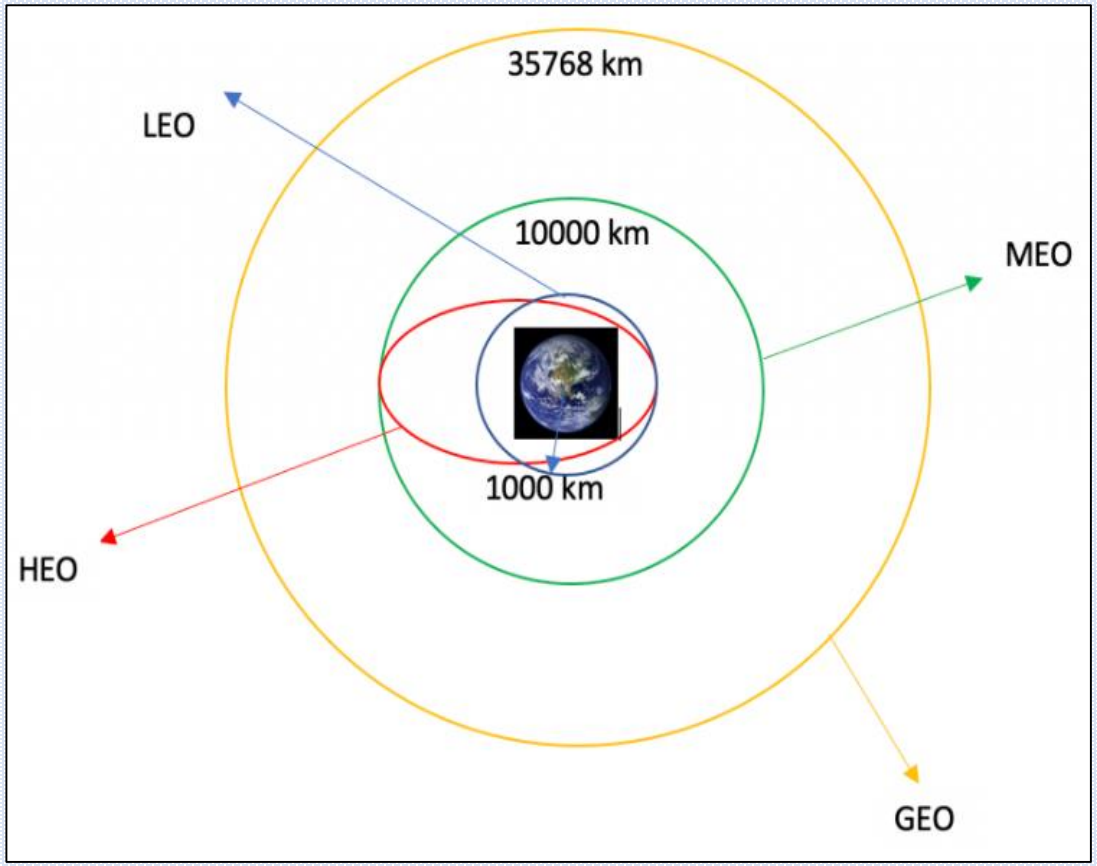
Data Collection - Scraping

- Web Scraping was applied on the Falcon 9 launch records with BeautifulSoup and parsed the table and converted it into a pandas dataframe.

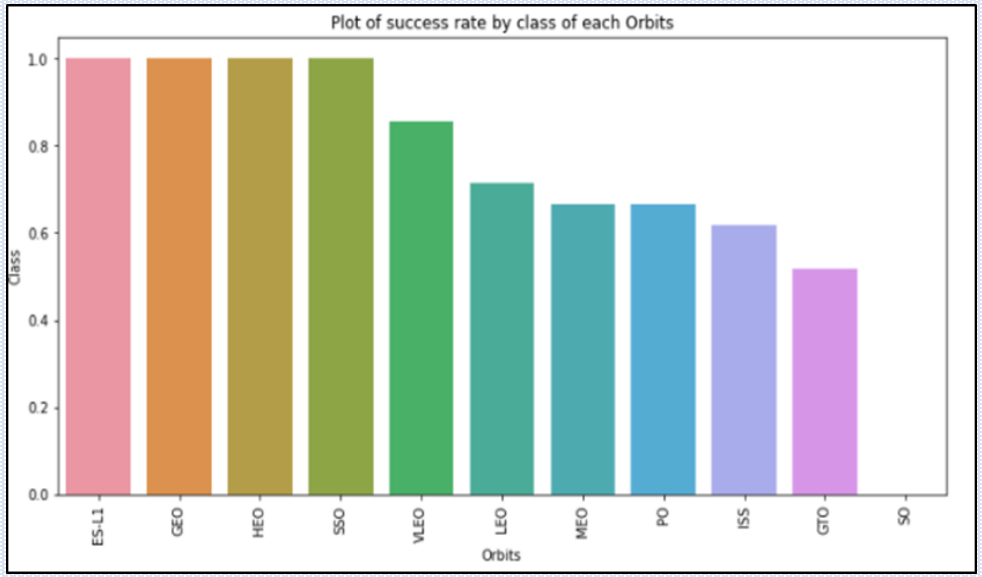
- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/ccb595081eeb4d8fe29c2c7ea7c252d962590712/Data%20Collection%20with%20Web%20Scraping.ipynb

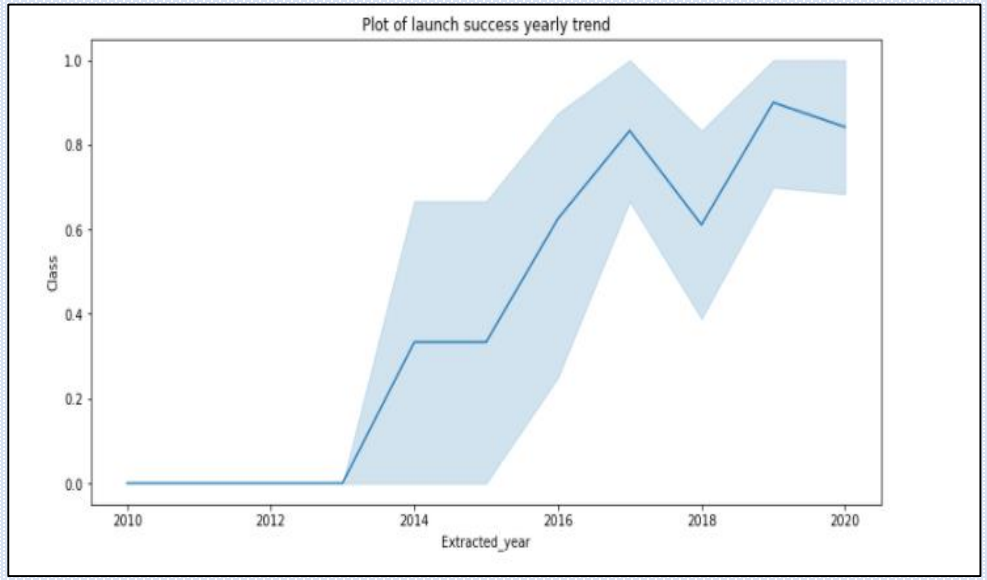
Data Wrangling

- Exploratory data analysis was conducted, and training labels were identified.
- The number of launches at each site were calculated, along with the number and occurrence of each orbits
- Landing outcome labels from outcome column were determined and the results were exported as a csv file.
- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/570ee98a01cd83cf10571b18fc1624acac55870e/EDA%20Lab%20.ipynb



EDA with Data Visualization





- Data was explored by Visualizing the Relationship Between Flight Number and Launch Site, Payload and Launch Site, Success rate of each orbit type, Flight Number and Orbit Type, the Launch Success Yearly Trend.

- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/01fe2c10cba010b196224d2d33b9f84c1c3d9676/EDA%20with%20Visualization%20lab.ipynb

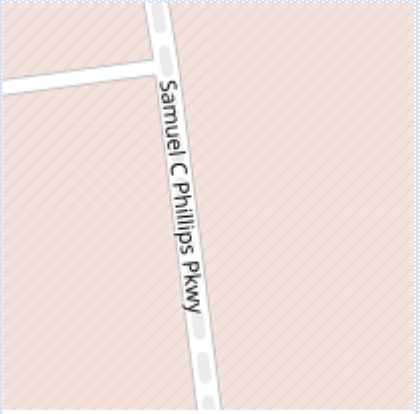
EDA with SQL

- The SpaceX dataset was loaded into a PostgreSQL database whilst Jupyter Notebook was in use.
- Exploratory Data Analysis with SQL was used to gain insight from the data with the objective of discovering:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/398b5afc67f74462f61c56bd26133d20e65f97aa/EDA%20with%20SQL%20lab%20-%20Space-X.ipynb

Build an Interactive Map with Folium

- Launch sites were marked, and map objects (markers, circles and lines) were used to mark the success or failure of launches for each site on the folium map.
- 0 is used for failure, and 1 for success.
- Using the color-labeled marker, cluster launch sites with high success rates could be identified.
- On the Folium Map:
  - Launch sites near railways, highways and coastlines could be identified

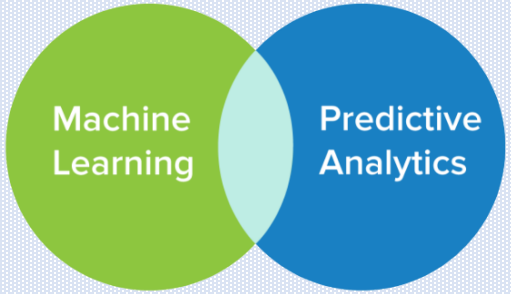


- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/398b5afc67f74462f61c56bd26133d20e65f97aa/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- The Interactive Dashboard is built with Plotly dash.
- Pie charts were built showing the total launches by a certain sites.
- Scatter Plot graph’s show the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/838d7eb9d2ebe5632da916a161ff3aeeef7709c7/dashboard.py

Predictive Analysis (Classification)

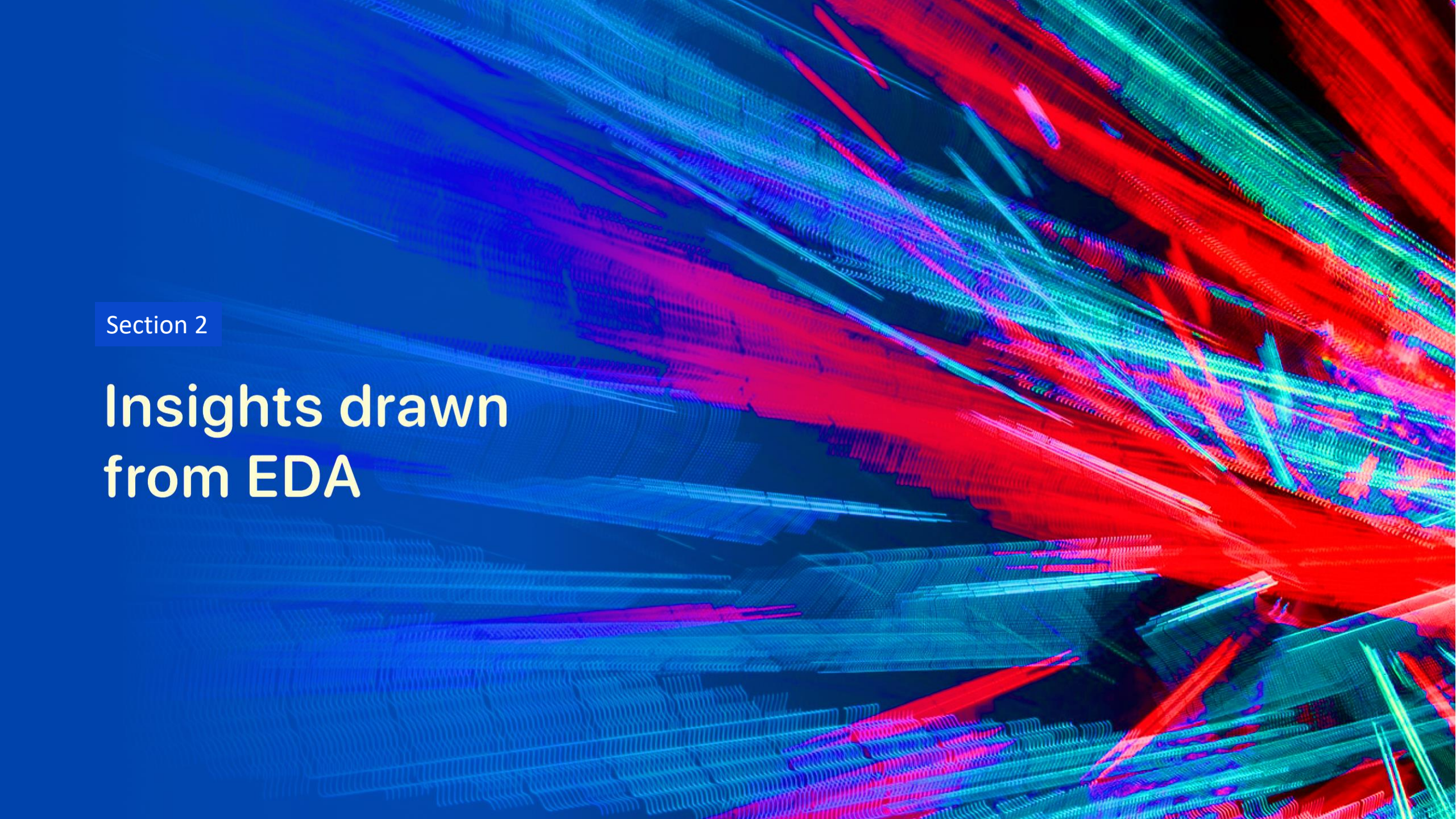


- Data was loaded using NumPy and Pandas, it was then transformed and split into training and testing categories.
- Models needed to be as accurate as possible and were improved upon using feature engineering and algorithm tuning.

- GitHub URL : https://github.com/MarphilMokoena/Space-X/blob/91a6660f1c6af964ce5d8836cb29667dfd3ebc00/Machine%20Learning%20Prediction.ipynb

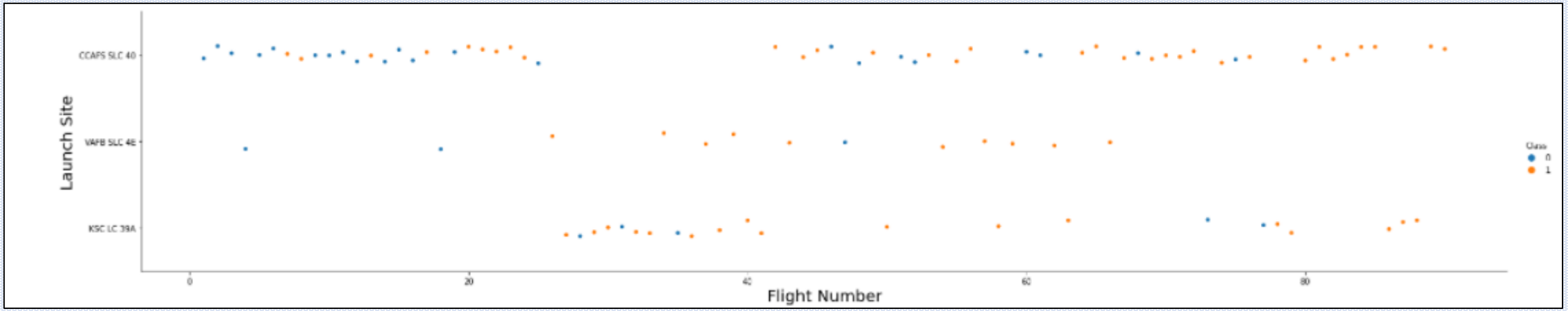
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

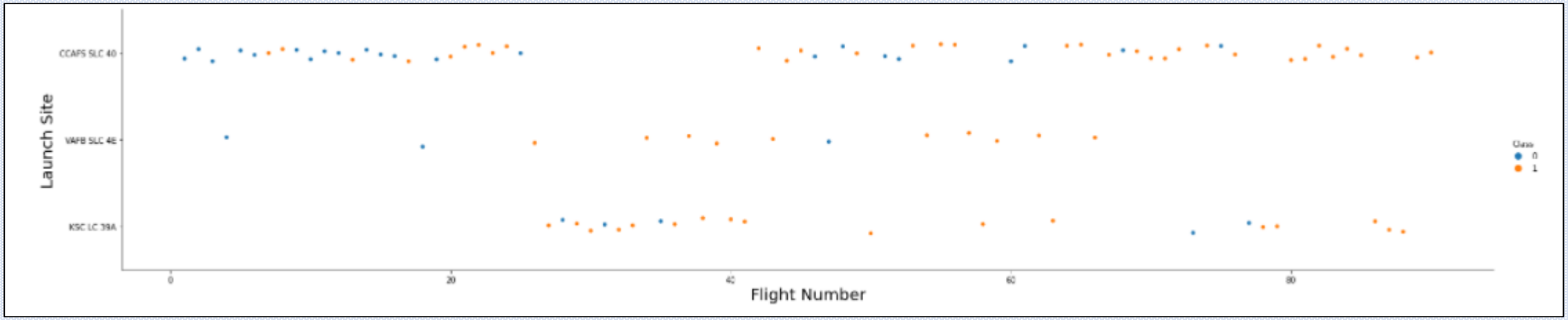
Flight Number vs. Launch Site

- The Larger the Flight Amount at a Launch Site, the Greater the Success Rate at a Launch Site.



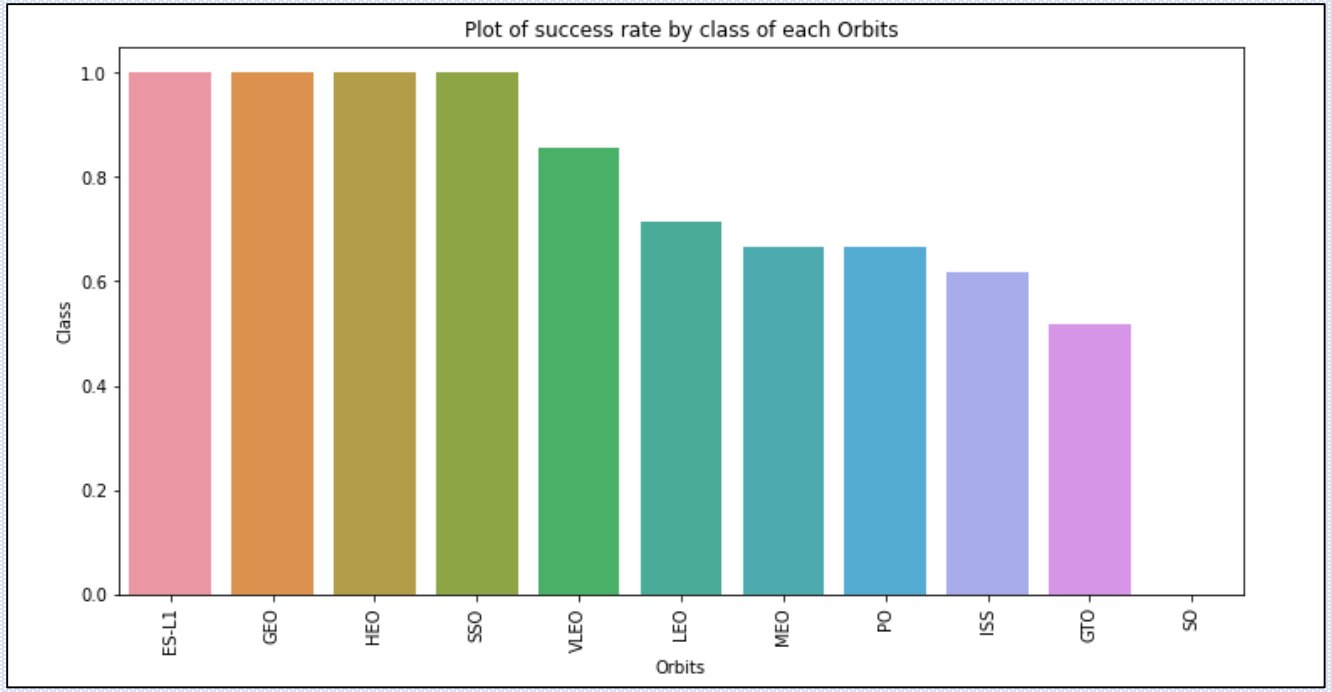
Payload vs. Launch Site

- The Greater the Payload Mass for Launch Site *CCAFS SLC 40* the Higher the Success Rate for the Rocket.



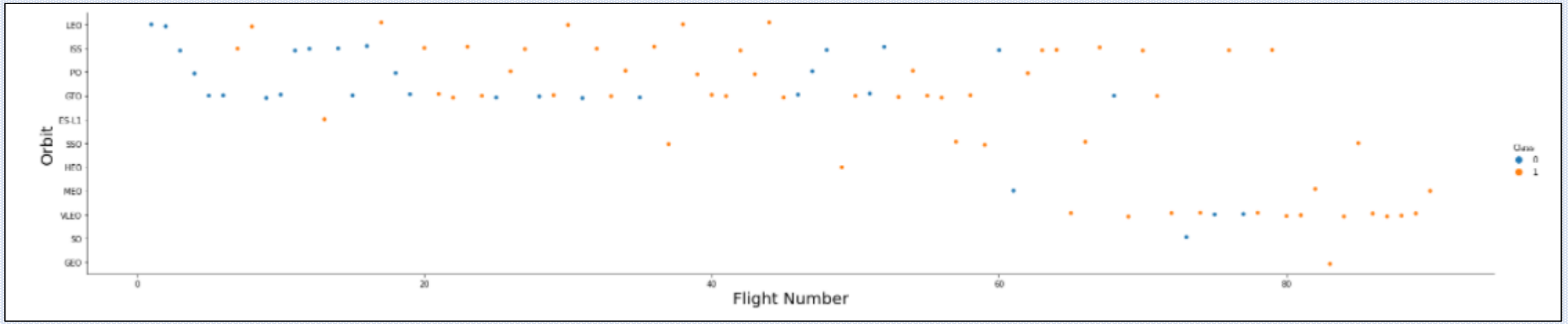
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO experienced the most success.
- Whilst SO experienced zero success
- LEO,MEO,PO,ISS,GTO ranged between 40-60% rate of success.



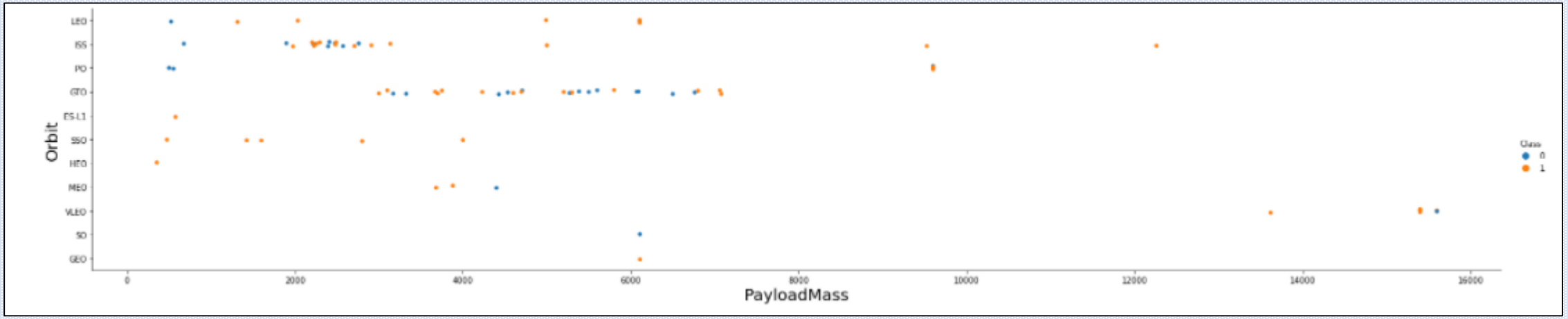
Flight Number vs. Orbit Type

- In the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



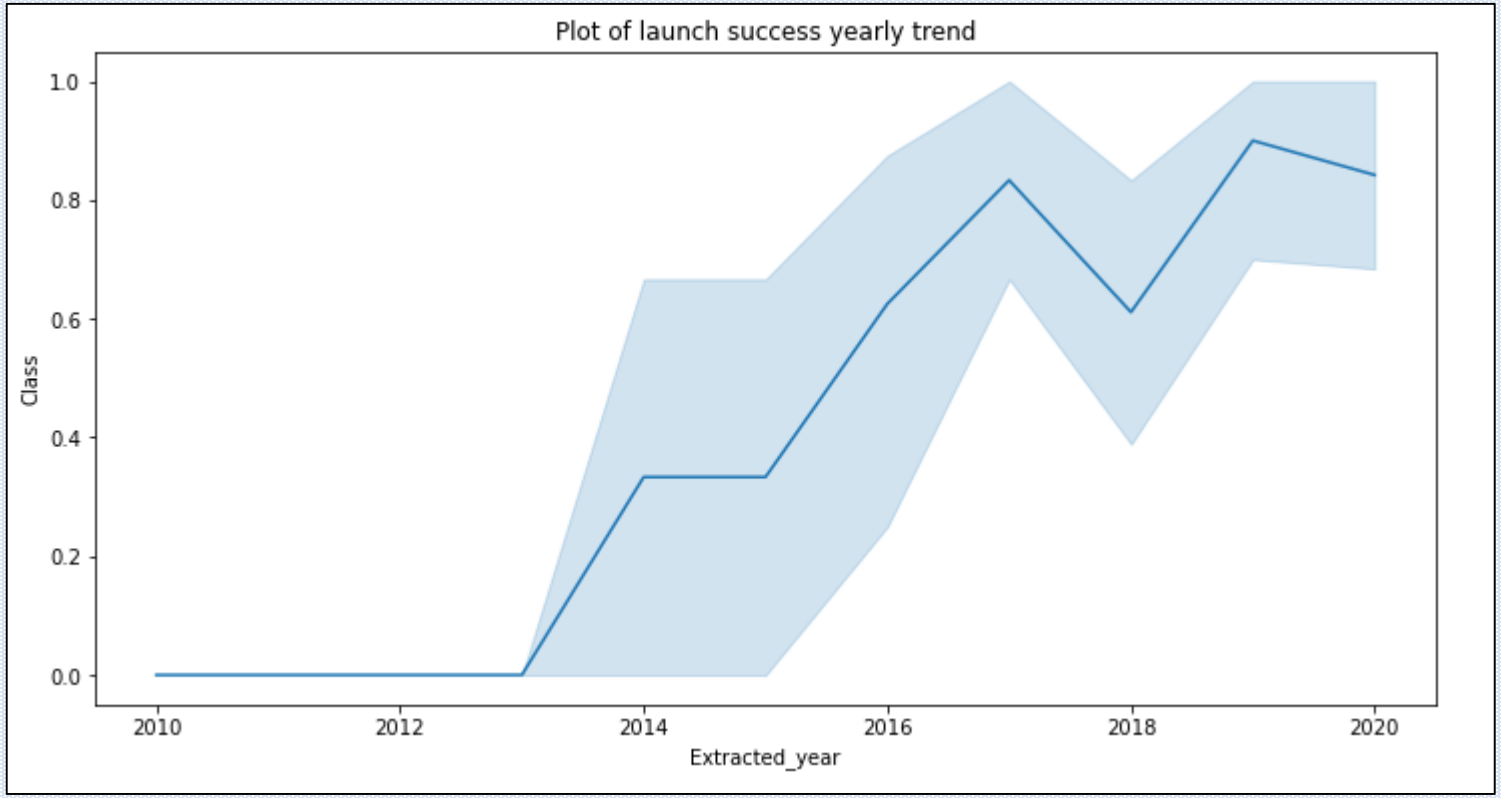
Payload vs. Orbit Type

- Heavy payloads influenced a higher rate of successful landing for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- The launch success rate has been increasing from 2013 till 2020 showing improvement.



All Launch Site Names

- The DISTINCT query was used to show only unique launch sites from the SpaceX data.

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Query results were limited to 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

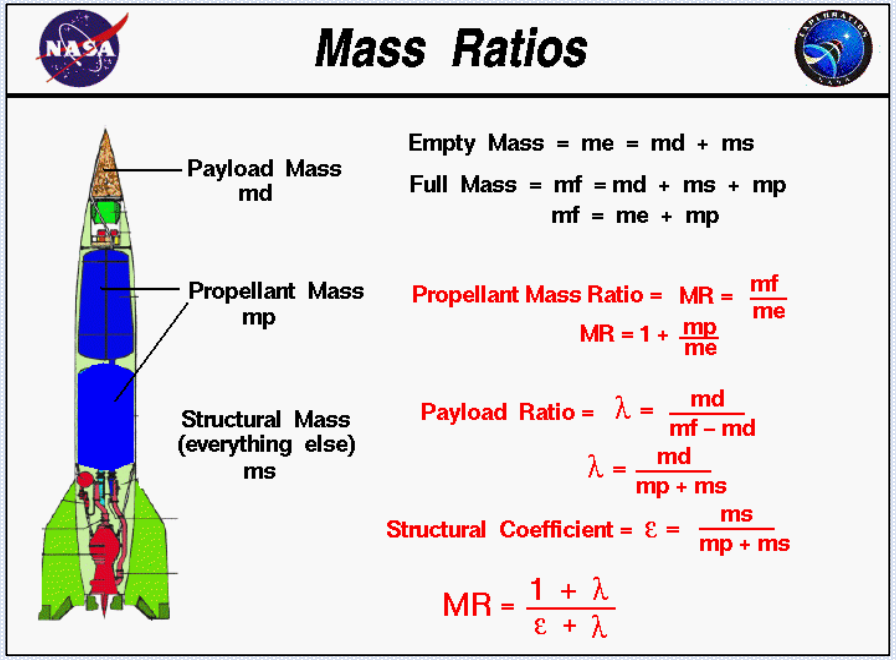
	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by boosters from NASA was 45,596 Kg and was revealed using the query below.

```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

	total_payloadmass
0	45596



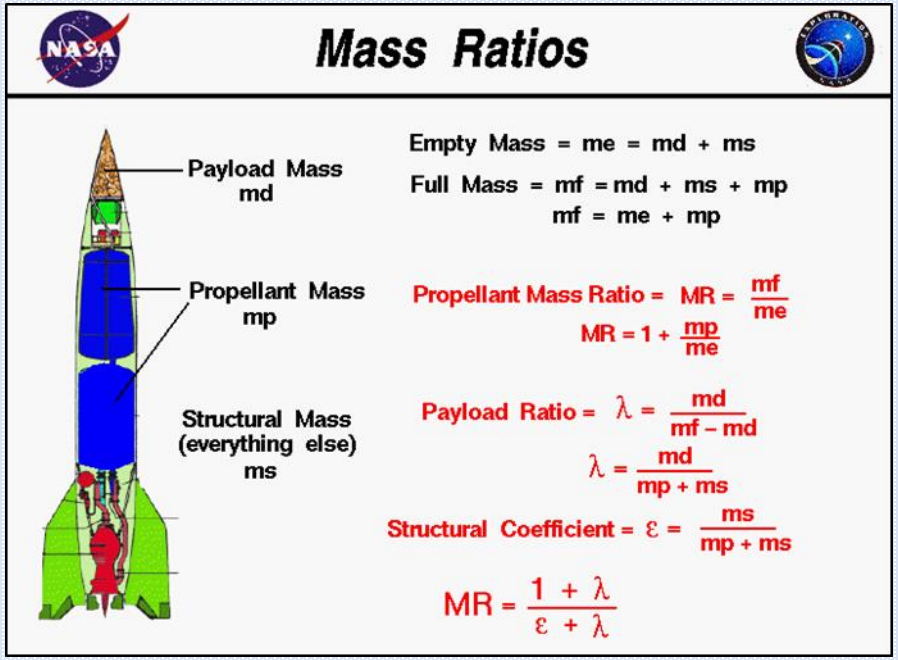
Average Payload Mass by F9 v1.1

- The Average payload mass carried by booster version F9 v1.1 as 2928.4 Kg.

Display average payload mass carried by booster version F9 v1.1

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

	avg_payloadmass
0	2928.4



First Successful Ground Landing Date

- The first successful landing was on the **22nd December 2015**

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

	firstsuccessfull_landing_date
0	2015-12-22

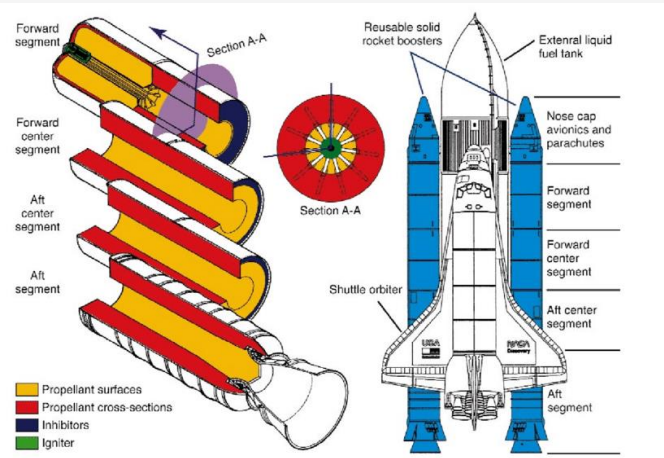
Successful Drone Ship Landing with Payload between 4000 and 6000

- The WHERE clause is used to filter for boosters which have successfully landed on the drone ship and applied on the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2



Total Number of Successful and Failure Mission Outcomes

- A wildcard like '%' was used to filter for WHERE MissionOutcome was a success (1) or a failure (0).

List the total number of successful and failure mission outcomes

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

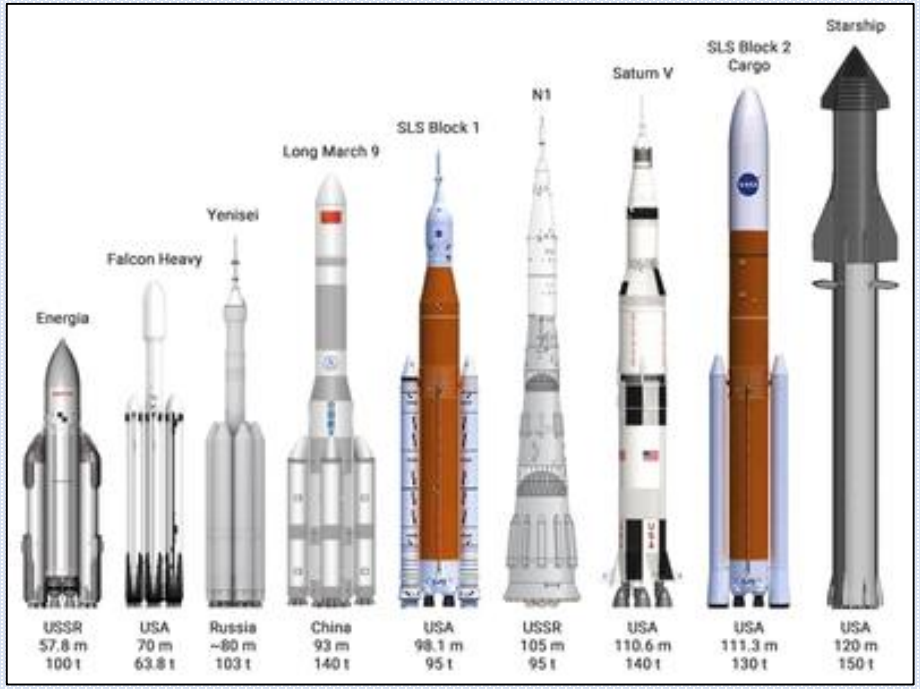
	successoutcome
0	100

```
The total number of failed mission outcome is:
```

	failureoutcome
0	1

Boosters Carried Maximum Payload

- The boosters that carried the maximum payload were identified using a subquery in the WHERE clause and the MAX() function.



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- Combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions are used to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Landing outcomes were selected and the COUNT of landing outcomes from the data. The WHERE clause was used to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- The GROUP BY clause is used to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

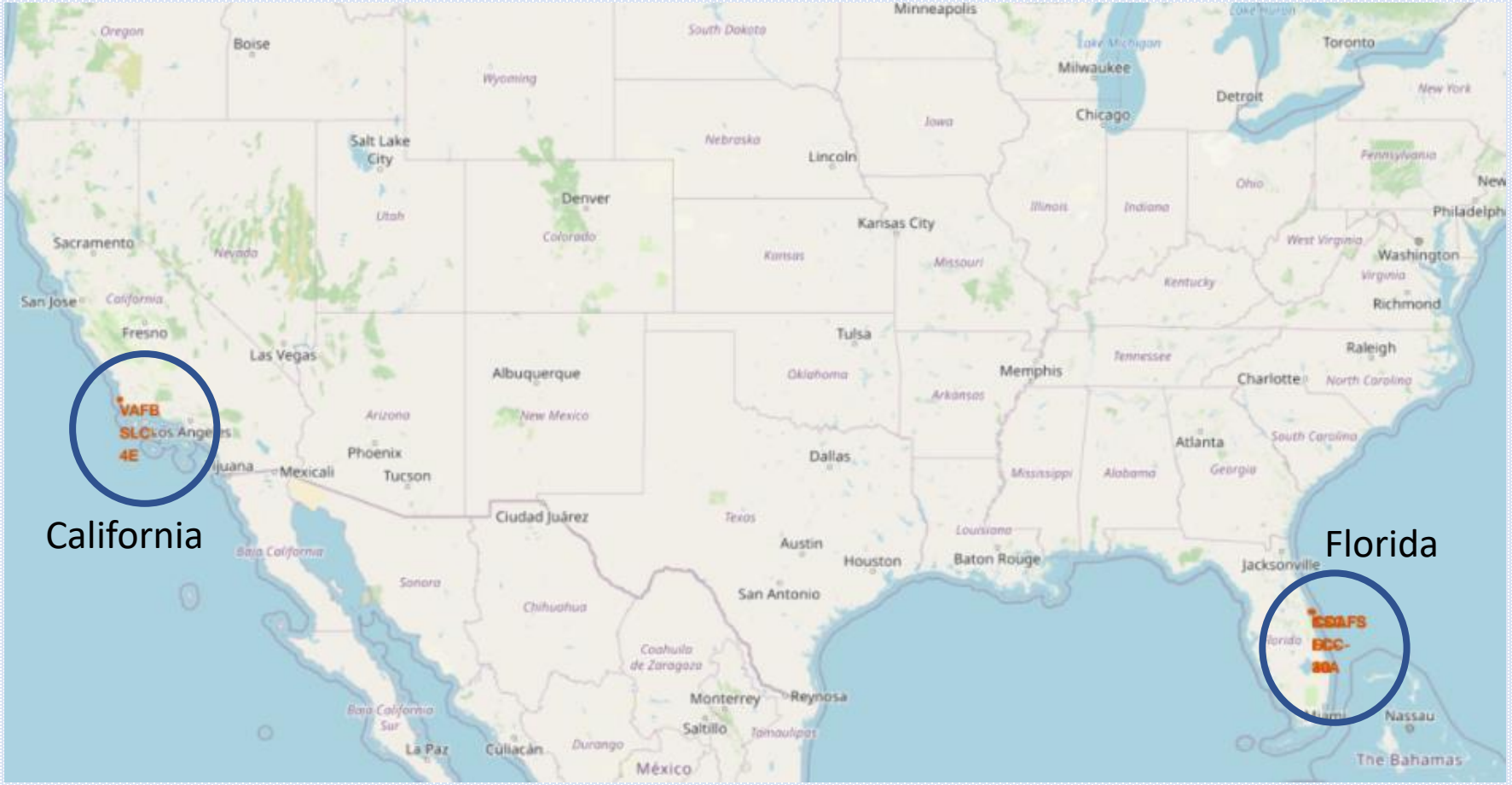
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

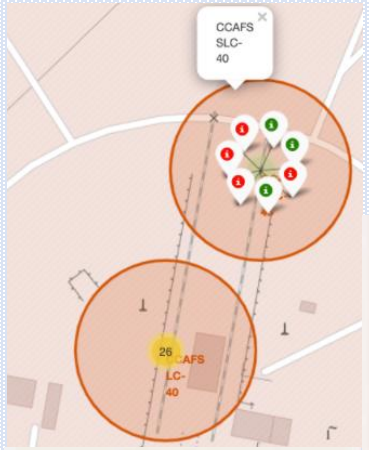
All Launch Sites from Folium Maps

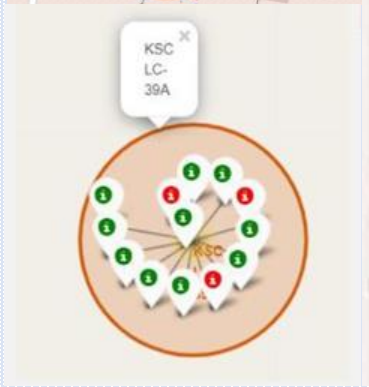
- **Space X Launch Sites are located in the United States and are coastal located in California (West) and Florida (East)**

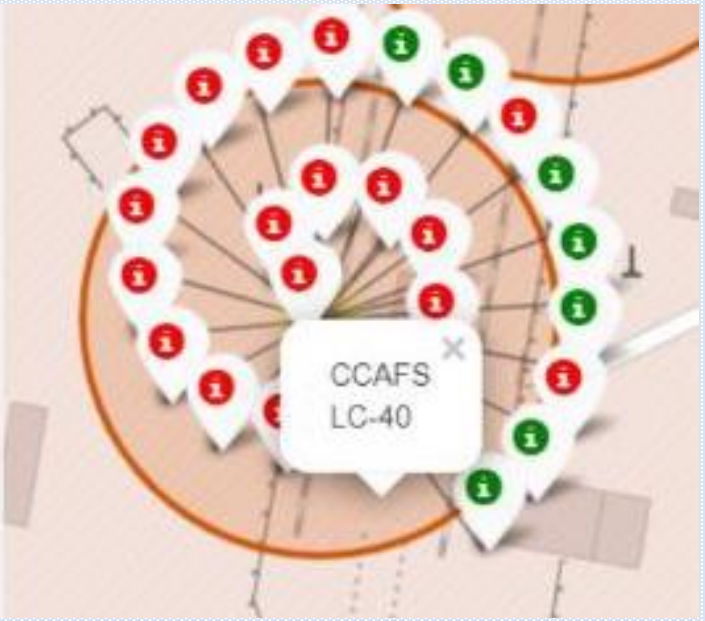


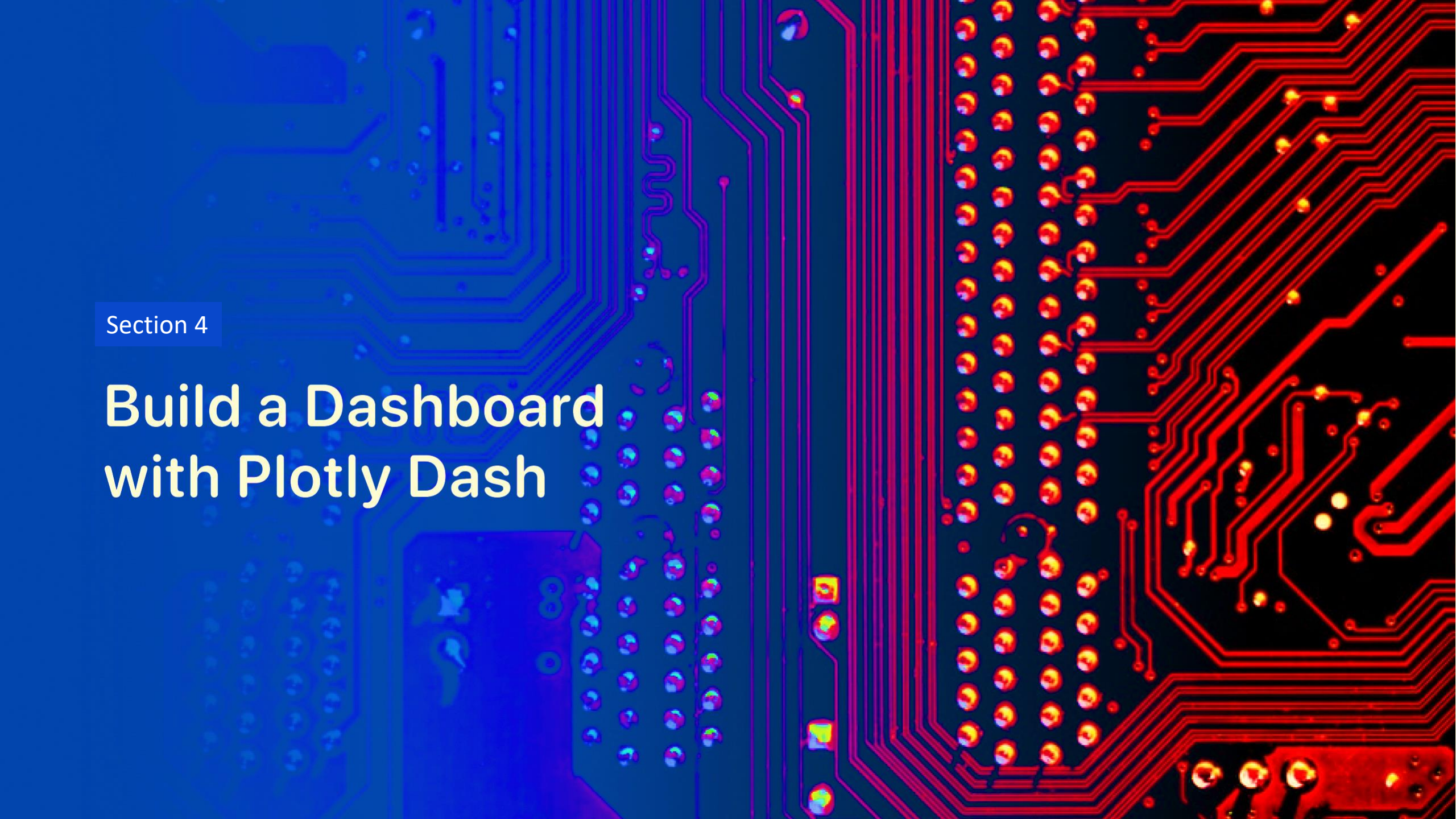
Color Labeled Launch Records

- The Green Marker is used to show Successful Launches and the Red Marker show Failure

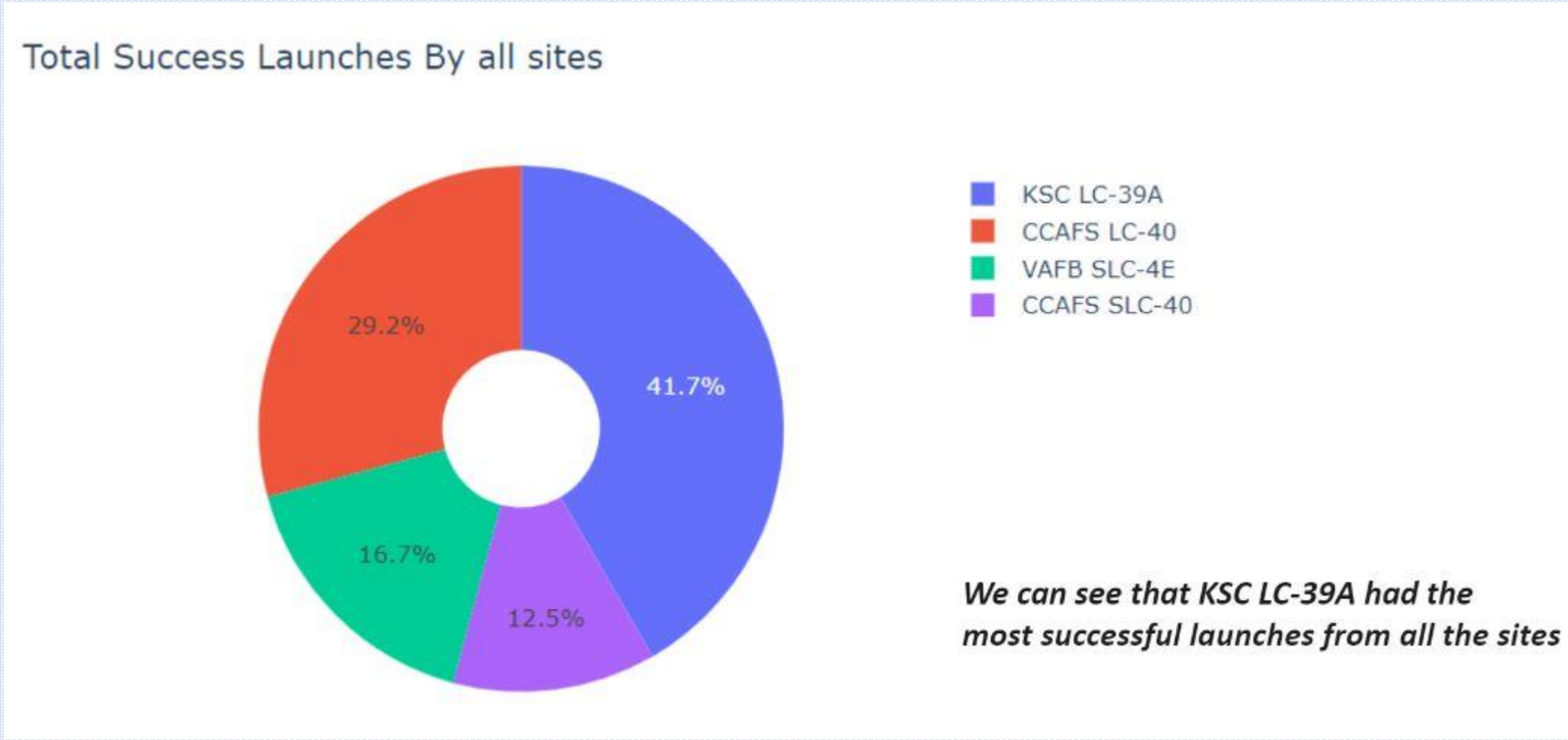






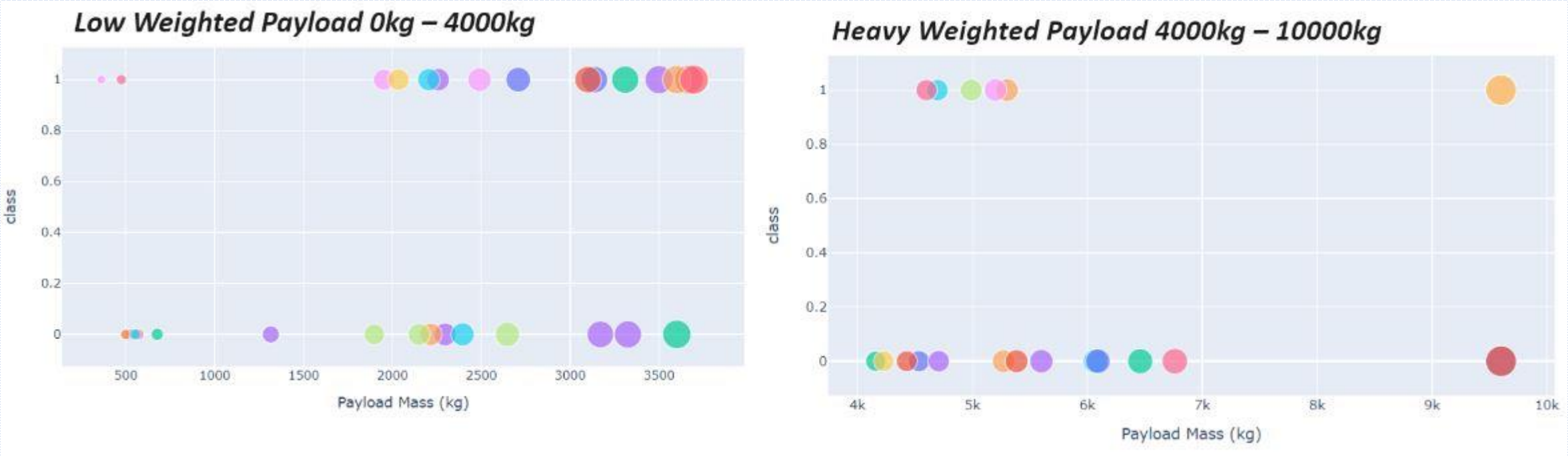

Section 4
Build a Dashboard with Plotly Dash

Pie chart Showing the Launch Success Count for all Sites



We can see that KSC LC-39A had the most successful launches from all the sites

Payload V Launch Outcome Scatter Plot for All Sites



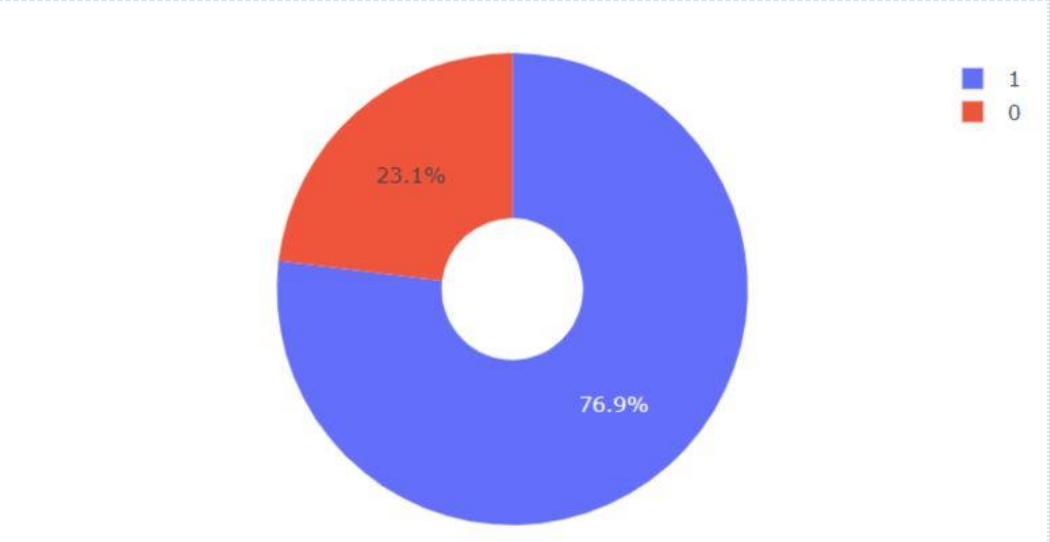
- The Success Rate for low Weighted Payloads is Higher than the Heavy Weighted Payloads

Launch Site with Highest Launch Success Ratio

❖ KSC LC-39A achieved a 76,9% Success rate and a 23,1% Failure rate

❖ Insights obtained from Dashboard:

- **KSC LC -39A** has the Highest Launch Success Rate
- **2000Kg – 10000Kg** is the payload range with the highest launch success rate
- **0Kg – 1000Kg** is the payload range with the lowest launch success rate
- **FT** F9 Booster version has the highest launch success rate



Total Success Launch for Site - KSC LC -39A

Section 5

Predictive Analysis (Classification)

Classification Accuracy

The Best Model is the **Decision Tree** with a **Score of 0,87**

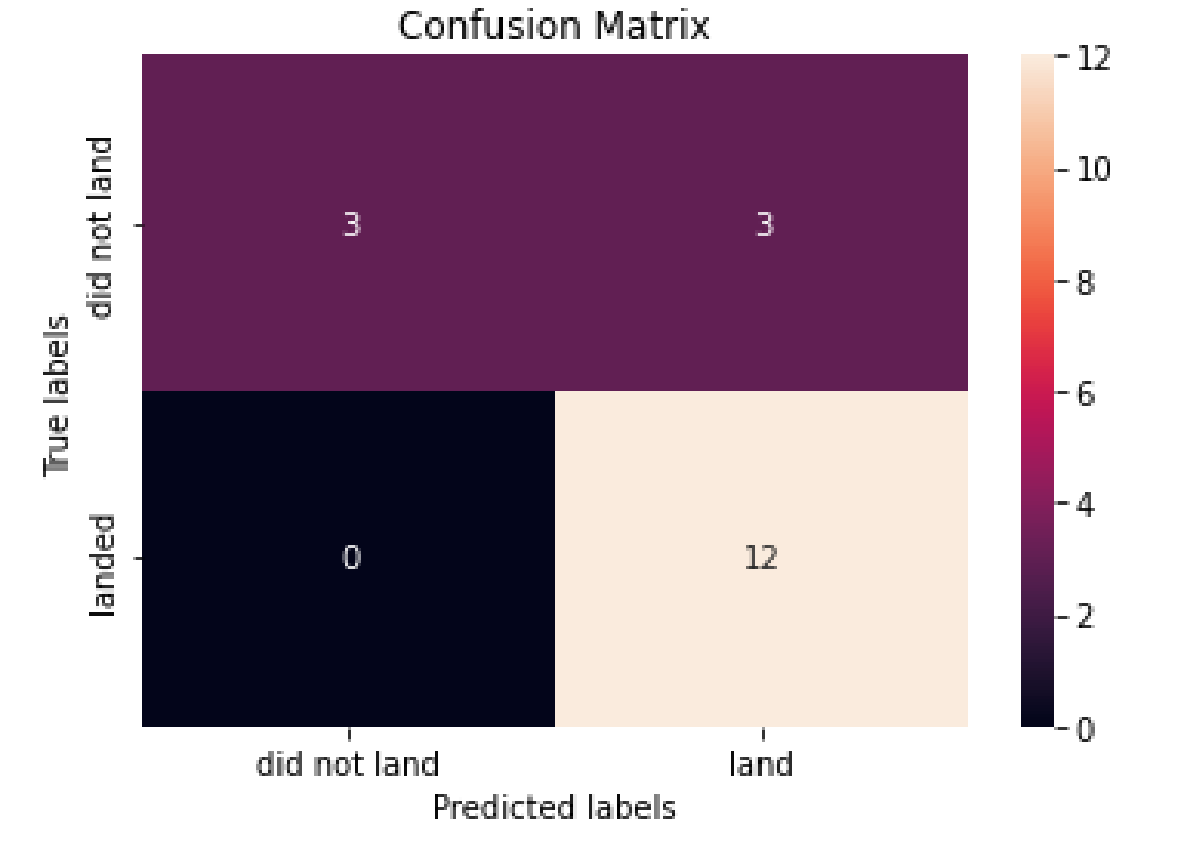
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

Decision Tree



❖ The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

❖ The biggest problem is false positives

Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!